NAG Toolbox for MATLAB

f11db

1 Purpose

f11db solves a system of linear equations involving the incomplete LU preconditioning matrix generated by f11da.

2 Syntax

```
[x, ifail] = f11db(trans, a, irow, icol, ipivp, ipivq, istr, idiag,
check, y, 'n', n, 'la', la)
```

3 Description

f11db solves a system of linear equations

$$Mx = y, \quad \text{or} \quad M^{\mathrm{T}}x = y,$$

according to the value of the parameter **trans**, where the matrix $M = PLDUQ$, corresponds to an incomplete LU decomposition of a sparse matrix stored in co-ordinate storage (CS) format (see Section 2.1.1 in the F11 Chapter Introduction), as generated by f11da.

In the above decomposition L is a lower triangular sparse matrix with unit diagonal elements, D is a diagonal matrix, U is an upper triangular sparse matrix with unit diagonal elements and, P and Q are permutation matrices. L, D and U are supplied to f11db through the matrix

$$C = L + D^{-1} + U - 2I$$

which is an **n** by **n** sparse matrix, stored in CS format, as returned by f11da. The permutation matrices P and Q are returned from f11da via the arrays **ipivp** and **ipivq**.

It is envisaged that a common use of f11db will be to carry out the preconditioning step required in the application of f11be to sparse linear systems. f11db is used for this purpose by the Black Box function f11dc.

f11db may also be used in combination with f11da to solve a sparse system of linear equations directly (see Section 8.5 of the document for f11da). This use of f11db is demonstrated in Section 9.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **trans – string**

Specifies whether or not the matrix M is transposed.

trans = 'N'

$Mx = y$ is solved.

trans = 'T'

$M^{\mathrm{T}}x = y$ is solved.

Constraint: **trans** = 'N' or 'T'.

2: **a**(**la**) – **double array**

The values returned in the array **a** by a previous call to f11da.

3: **irow**(**la**) – **int32 array**
4: **icol**(**la**) – **int32 array**
5: **ipivp**(**n**) – **int32 array**
6: **ipivq**(**n**) – **int32 array**
7: **istr**(**n** + **1**) – **int32 array**
8: **idiag**(**n**) – **int32 array**

The values returned in arrays **irow**, **icol**, **ipivp**, **ipivq**, **istr** and **idiag** by a previous call to f11da.

9: **check** – **string**

Specifies whether or not the CS representation of the matrix M should be checked.

check = 'C'

Checks are carried on the values of **n**, **irow**, **icol**, **ipivp**, **ipivq**, **istr** and **idiag**.

check = 'N'

None of these checks are carried out.

See also Section 8.2.

Constraint: **check** = 'C' or 'N'.

10: **y**(**n**) – **double array**

The right-hand side vector y.

5.2 Optional Input Parameters

1: **n** – **int32 scalar**

n, the order of the matrix M. This **must** be the same value as was supplied in the preceding call to f11da.

Constraint: **n** ≥ 1.

2: **la** – **int32 scalar**

Default: The dimension of the arrays **a**, **irow**, **icol**. (An error is raised if these dimensions are not equal.)

This **must** be the same value as was supplied in the preceding call to f11da.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **x**(**n**) – **double array**

The solution vector x.

2: **ifail** – **int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **trans** ≠ 'N' or 'T',
or **check** ≠ 'C' or 'N'.

ifail = 2

On entry, **n** < 1.

ifail = 3

On entry, the CS representation of the preconditioning matrix M is invalid. Further details are given in the error message. Check that the call to f11db has been preceded by a valid call to f11da and that the arrays **a**, **irow**, **icol**, **ipivp**, **ipivq**, **istr** and **idiag** have not been corrupted between the two calls.

7 Accuracy

If **trans** = 'N' the computed solution x is the exact solution of a perturbed system of equations $(M + \delta M)x = y$, where

$$|\delta M| \leq c(n)\epsilon P|L||D||U|Q,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***. An equivalent result holds when **trans** = 'T'.

8 Further Comments

8.1 Timing

The time taken for a call to f11db is proportional to the value of **nnzc** returned from f11da.

8.2 Use of check

It is expected that a common use of f11db will be to carry out the preconditioning step required in the application of f11be to sparse linear systems. In this situation f11db is likely to be called many times with the same matrix M. In the interests of both reliability and efficiency, you are recommended to set **check** to 'C' for the first of such calls, and to 'N' for all subsequent calls.

9 Example

```
trans = 'N';
a = [1;
     1;
     -1;
     2;
     2;
     3;
     -2;
     1;
     -2;
     1;
     1;
     1;
     1;
     0.3333333333333333;
     -0.6666666666666666;
     -0.3333333333333333;
```

```
     0.5;
     0.6666666666666667;
     -2;
     0.3333333333333333;
     1.5;
     -2.999999999999996;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0];
irow = [int32(1);
     int32(1);
     int32(2);
     int32(2);
     int32(2);
     int32(3);
     int32(3);
     int32(4);
     int32(4);
     int32(4);
     int32(4);
     int32(1);
     int32(1);
     int32(2);
     int32(2);
     int32(3);
     int32(3);
     int32(3);
     int32(4);
     int32(4);
     int32(4);
     int32(4);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
     int32(0)];
icol = [int32(2);
     int32(3);
     int32(1);
     int32(3);
     int32(4);
     int32(1);
     int32(4);
     int32(1);
     int32(2);
     int32(3);
     int32(4);
     int32(1);
     int32(3);
     int32(2);
     int32(4);
     int32(2);
     int32(3);
     int32(4);
     int32(1);
     int32(2);
     int32(3);
     int32(4);
     int32(0);
     int32(0);
     int32(0);
     int32(0);
```

```
     int32(0);
     int32(0);
     int32(0);
     int32(0)];
ipivp = [int32(1);
     int32(3);
     int32(2);
     int32(4)];
ipivq = [int32(2);
     int32(1);
     int32(3);
     int32(4)];
istr = [int32(12);
     int32(14);
     int32(16);
     int32(19);
     int32(23)];
idiag = [int32(12);
     int32(14);
     int32(17);
     int32(22)];
check = 'C';
y = [5;
     13;
     -5;
     4];
[x, ifail] = f11db(trans, a, irow, icol, ipivp, ipivq, istr, idiag,
check, y)
```

```
x =
    1.0000
    2.0000
    3.0000
    4.0000
ifail =
           0
```